Q3. Code Review: User List Screen Bugs

1. Null Safety Violation

Issue: *selectedUser* is declared but not initialized, leading to potential null reference errors when accessing *selectedUser.toUpperCase()*

Effect: Will throw **Null check operator used on a null value** runtime error when tapping the selected user display before selecting any user

Fix:

```
String selectedUser = ''; // Initialize with empty string

// OR

String? selectedUser; // Make nullable and handle null case
```

2. ListView Constraints

Issue: *ListView.builder* is placed inside a *Column* without constraints

Effect: Will throw **Vertical viewport was given unbounded height** error

Fix: Wrap ListView with Expanded:

```
Expanded(
 child: ListView.builder(
  // ...
 ),
)
```

3. Search Functionality Bug

Issue: *searchUsers()* modifies the original users list permanently

Effect: After searching, original list is lost and can't be restored

Fix: Maintain separate lists for original and filtered users:

```
List<String> _allUsers = ['Alice', 'Bob', 'Charlie', 'Diana'];

List<String> users = [];

@override

void initState() {
```

```
 super.initState();
 users = _allUsers;
}

void searchUsers(String query) {
 setState(() {
  users = query.isEmpty
     ? _allUsers
     : _allUsers.where((user) =>
user.toLowerCase().contains(query.toLowerCase())).toList();
 });
}
```

4. Delete Operation Bug

Issue: Deleting users modifies the displayed list but not the original list

Effect: After search, deleted users reappear

Fix: Modify both lists:

```
onPressed: () {
 _allUsers.removeAt(index);
 users.removeAt(index);
 setState(() {});
},
```

5. Add User Inconsistency

Issue: *addUser()* only adds to the displayed list

Effect: New users disappear after search

Fix: Add to both lists:

```
void addUser() {
  final newUser = 'New User ${_allUsers.length + 1}';
  _allUsers.add(newUser);
  users.add(newUser);
  setState(() {});
}
```

6. Case Sensitivity in Search

Issue: Search is case-sensitive in the current implementation

Effect: *alice* won't match *Alice*

Fix: Already handled in current search implementation with *toLowerCase()*

7. Missing Key for List Items

Issue: List items don't have keys

Effect: Potential rendering issues during updates

Fix: Add keys to ListTile:

```
ListTile(
  key: ValueKey(users[index]),
  // ... rest of the code
)
```

8. No Empty State Handling

Issue: No handling when user list is empty

Effect: Shows empty space when no users exist

Fix: Add empty state widget:

```
itemCount: users.isEmpty ? 1 : users.length,
itemBuilder: (context, index) {
  if (users.isEmpty) {
```

```
    return Center(child: Text('No users found'));
  }
  // ... existing ListTile code
}
```

9. FloatingActionButton Placement

Issue: FAB is inside Column without proper positioning

Effect: Appears in middle of content

Fix: Move to Scaffold's floatingActionButton property:

```
return Scaffold(
  floatingActionButton: FloatingActionButton(
    onPressed: addUser,
    child: Icon(Icons.add),
  ),
  // ... rest of scaffold
);
```

10. State Management Issue

Issue: Direct list modification with setState

Effect: Potential performance issues and harder debugging

Fix: Create copy before modification:

```
void addUser() {
  final newUsers = List<String>.from(_allUsers);
  newUsers.add('New User ${newUsers.length + 1}');
  setState(() {
    _allUsers = newUsers;
    users = newUsers;
  });
}
```